

REMARKS

The Office Action mailed September 17, 1998, has been carefully considered.

In the Office Action, the Examiner objected to the specification on the basis that the serial number for the Waldo, et al., application incorporated by reference was not provided. In addition, the Examiner rejected claims 31 through 33 under 35 U. S. C. §103 as being obvious over M. Betz, "Interoperable objects: laying the foundation for distributed object computing" (hereinafter "Betz") in view of Orbix, "The Orbix Architecture" (hereinafter "Orbix"), claims 1, 3, 4, 7-11, 13, 14, 17-21, 23, 34, and 27-30 under 35 U. S. C. §103 as being obvious over W. Rosenberry, et al., Understanding DCE, chapters 1-3 and 6 (hereinafter "Rosenberry") in view of Orbix, and claims 2, 5, 6, 12, 15, 16, 22, 25 and 26 under 35 U. S. C. §103 as being obvious over Rosenberry in view of Orbix and J. Mitchell, et al., "An overview of the Spring System," (hereinafter Mitchell). No claims have been allowed.

By this amendment, Applicants are amending the specification to provide the serial number for the Waldo, et al., patent application. In addition, since that application has issued as a patent, the specification is further being amended to provide the patent number.

Applicants respectfully submit that the claims patentably distinguish over the references.

Addressing first claim 1, that claim is directed to a stub retrieval and loading subsystem for use in connection with a remote method invocation system. The stub retrieval and loading subsystem controls the retrieval and loading of a stub for a remote method into an execution environment to facilitate invocation of the remote method by a program executing in the execution environment, the stub retrieval subsystem comprises a stub retriever and a stub loader. The stub retriever initiates the retrieval of the stub from a server associated with processing of the remote method. The stub loader, when the stub is received by the stub retriever, loads the stub into the execution environment, thereby to make the stub available for use in remote invocation of the remote method.

Raspberry describes a distributed computing environment (DCE) which can be provided by a number of interconnected computer systems. In particular connection with claim 1, and as specifically cited by the Examiner, section 3.1.2 describes the use of a remote procedure call (RPC), in which a "client" operating in one process can, by use of a call to stub code provided in that process, initiate processing by a "server" operating in another process, which may be operating in a different computer system. As described in section 3.1.2, the stub code takes the place of local procedures which the client is expecting, and essentially packages the call, sends it using an appropriate communications methodology to the other process. The server processes the request and generates results, which are sent back to the stub code. The stub code, in turn, receives the results from the server and unpackages them and provides them to the client.

While Raspberry describes stub code and its use in connection with an RPC, Applicants respectfully submit that it does not suggest the invention recited in claim 1. The invention recited in claim 1 is directed to an arrangement in which stub code, which does not initially form part of the execution environment (for example, a process) for a client, is retrieved and loaded into the execution environment. In Raspberry, there is no suggestion of initiating the retrieval of stub code from a server associated with processing of the remote method, or of loading the retrieved stub code into an execution environment after the stub code has been received, both of which are called for in claim 1. Raspberry at most suggests that stub code is provided as part of the process when the process is created. Accordingly, Applicants respectfully submit that Raspberry neither teaches nor suggests the arrangement recited in claim 1.

The Examiner cites Orbix as teaching dynamic loading of stub code, referencing the Dynamic Invocation Interface mentioned on page 18. While Orbix does suggest that stub code can be loaded dynamically, there is no suggestion where the stub code that is to be dynamically linked is obtained from, more specifically there is no suggestion that the stub code be provided by a server that is associated with processing of the remote method. As noted above, a benefit of the invention is that, by having the stub obtained from the server providing the remote method or procedure, the server can ensure that the stub can exactly define the invocation requirements of the remote method or procedure, in which case run-time errors and inefficiencies which may result from mismatches

between the stub that is used and the requirements of the remote method or procedure that is invoked can be minimized. This is nowhere suggested in Orbix. And further, there is no suggestion in Orbix that the stub code be provided by the server associated with processing of the remote method.

Applicants have reviewed the other references and submit that they, whether considered individually or in combination, also neither teach nor suggest the invention recited in claim 1.

Applicants further submit that independent method claim 11 and independent computer program product claim 21 also distinguish over Raspberry for the reasons set forth above in connection with claim 1. Claim 11 is directed to a method comprising steps along the lines of those performed by the apparatus recited in claim 1. Claim 21 is directed to a computer program product with code devices for enabling a computer to perform operations along the lines of those performed by the apparatus recited in claim 1. Applicants further submit that claims 2-10, 12-20 and 22-30 are allowable for depending from allowable independent claims.

Accordingly, Applicants respectfully submit that the claims 1-30 patentably distinguish over Raspberry and Orbix, whether considered individually or in combination.

Applicants further submit that claims 31-33 patentably distinguish over Betz and Orbix, whether considered individually or in combination. Claim 31 is directed to a stub retrieval and loading subsystem for use in connection with a remote method invocation system. The stub retrieval and loading subsystem controls the retrieval and loading of a stub for a remote method into an execution environment to facilitate invocation of the remote method by a program executing in the execution environment. The stub retrieval subsystem comprises a computer and a control arrangement for controlling the computer. The control arrangement is recited as comprising a stub retrieval module and a stub loader module. The stub retrieval module controls the computer to initiate a retrieval of the stub from a server that is associated with processing of the remote method. The stub loader module controls the computer to, when the stub is received in response to the stub retrieval module, load the stub into the execution environment, thereby to make the stub available for use in remote invocation of the remote method.

Claim 32 is directed to a control arrangement along the lines of that recited in claim 31.

Finally, claim 33 is directed to a system for distributing code stored on a computer readable medium and executable by a computer. The code includes a plurality of modules each configured to control the computer to facilitate the retrieval and loading of a stub for a remote method into an execution environment to facilitate invocation of the remote method by a program executing in said execution environment. The system is recited as including a stub retrieval module and a stub loader module along the line of those directed in claim 33.

Betz describes the use of "inter-operable" objects in a computer, which allows a process in one address space to request the services of an object in another address space, or which allows processes in separate address spaces share an object in a third address space (see the second full paragraph on page 4 of the copy of Betz provided with the Office Action). Betz describes a number of inter-operable object models, which have been proposed or used in products provided by a number of vendors. However as with Raspberry, there is no suggestion in Betz of initiating the retrieval of stub code or of loading the retrieved stub code into an execution environment after the stub code has been received, both of which are called for in claims 31-33. Accordingly, Applicants respectfully submit that Betz neither teaches nor suggests the arrangement recited in claim 1.

The Examiner cites Orbix as teaching dynamic loading of stub code, referencing the Dynamic Invocation Interface mentioned on page 18. While Orbix does suggest that stub code can be loaded dynamically, there is no suggestion where the stub code that is to be dynamically linked is obtained from, more specifically there is no suggestion that the stub code be provided by a server that is associated with processing of the remote method. As noted above, a benefit of the invention is that, by having the stub obtained from the server providing the remote method or procedure, the server can ensure that the stub can exactly define the invocation requirements of the remote method or procedure, in which case run-time errors and inefficiencies which may result from mismatches between the stub that is used and the requirements of the remote method or procedure that is invoked can be minimized. This is nowhere suggested in Orbix. And further, there is no suggestion in Orbix that the stub code be provided by the server associated with processing of the remote method.

Accordingly, Applicants respectfully submit that claims 31-33 patentably distinguish over Betz and Orbix, whether considered individually or in combination.

Applicants have also reviewed the other references cited in the Office Action, and respectfully submit that the claims patentably distinguish thereover.

In view of the above, Applicants respectfully submit that the claims patentably distinguish over the references.

It is believed that this application is allowable, and a notice of allowability is respectfully solicited.

Respectfully submitted,



Richard A. Jordan

Reg. No. 27,807

Telephone Numbers

Voice +1 (781) 431-1357

Facsimile +1 (781) 235-8326